AFRL-RI-RS-TR-2010-150

NEVER-ENDING LEARNING

Carnegie Mellon University

August 2010

Sponsored By
Defense Advanced Research Projects Agency
Darpa Order No. AR75/00

FINAL TECHNICAL REPORT

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the 88th ABW, Wright-Patterson AFB Public Affairs Office and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RI-RS-TR-2010-150 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/s/
DEBORAH A. CERINO
Work Unit Manager

/s/
MICHAEL WESSING, Deputy Chief
For
JOSEPH CAMERA, Chief
Information & Intelligence Exploitation Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
AUGUST 2010	Final	September 2008 – February 2010

4. TITLE AND SUBTITLE

NEVER-ENDING LEARNING

5a. CONTRACT NUMBER
N/A

5b. GRANT NUMBER
FA8750-08-1-0009

5c. PROGRAM ELEMENT NUMBER
62304E

6. AUTHOR(S)

Tom Mitchell

5d. PROJECT NUMBER
TRLG

5e. TASK NUMBER
00

5f. WORK UNIT NUMBER
03

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Carnegie Mellon University
5000 Forbes Avenue
Pittsburg, PA 15213-3815

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Defense Advanced Research Projects Agency
3701 North Fairfax Drive
Arlington, VA 22203-1714

AFRL/RIED
525 Brooks Road
Rome, NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)
N/A

11. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-RI-RS-TR-2010-150

12. DISTRIBUTION AVAILABILITY STATEMENT
Approved For Public Release; Distribution Unlimited. PA# 88ABW-2010-4123
Date Cleared: 2-August-2010

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The goal of this research was to explore a new approach to machine learning, called Never-Ending Learning. Although machine learning research has been increasingly successful in recent years, this effort addressed developing a machine learning system that learns cumulatively forever, using what was learned yesterday to improve its ability to learn tomorrow, and improving daily, indefinitely. The thesis underlying this research is that the vast redundancy of information on the web (e.g., many facts are stated multiple times in different ways) will enable a system with the right learning mechanisms and capabilities for self-reflection to learn with only occasional outside supervision. A general approach is described for building a never-ending language learner that uses semi-supervised learning methods, an ensemble of varied knowledge extraction methods, and a flexible knowledge base representation that allows the integration of the outputs of those methods.

15. SUBJECT TERMS

Machine Learning, Semi-Supervised Learning, Information Extraction

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Deborah A. Cerino
U	U	U	UU	24	19b. TELEPHONE NUMBER (*Include area code*) N/A

Table of Contents

List of Figures

List of Tables

1. Introduction

The goal of this research was to explore a new approach to machine learning, called Never Ending Learning. Although machine learning research has been increasingly successful in recent years, this effort addressed developing a machine learning system that learns cumulatively forever, using what was learned yesterday to improve its ability to learn tomorrow, and improving daily, indefinitely. This effort performed research toward such a system, including the development of a system capable of operating continuously, 24x7, and improving its competence daily for at least a month before hitting a possible plateau in performance."

Progress toward the longer-term goal of producing a never-ending language learner, is described herein. By a "never-ending language learner" we mean a computer system that runs 24 hours per day, 7 days per week, forever performing two tasks each day:

1. Reading task: extracting information from web text to further populate a growing knowledge base of structured facts and knowledge.

2. Learning task: learning to read better each day than the day before, as evidenced by its ability to go back to yesterday's text sources and extract more information more accurately.

The thesis underlying this research is that the vast redundancy of information on the web (e.g., many facts are stated multiple times in different ways) will enable a system with the right learning mechanisms and capabilities for self-reflection to learn with only occasional outside supervision. We also hypothesize that the architecture we describe here can be used as a platform for conducting increasingly sophisticated research in natural language understanding.

We first describe a general approach to building a never-ending language learner that uses semi-supervised learning methods, an ensemble of varied knowledge extraction methods, and a flexible knowledge base representation that allows the integration of the outputs of those methods. We also discuss design principles for implementing this approach.

We then describe a prototype implementation of our approach, called Never-Ending Language Learner (NELL). At present, NELL acquires two types of knowledge: (1) knowledge about what noun phrases refer to some specified semantic categories, such as cities, companies, and universities, and (2) knowledge about what pairs of noun phrases satisfy some specified semantic relations, such as hasOfficesIn (organization, location). NELL learns to acquire these two types of knowledge in a variety of ways. It learns free-form text

patterns for extracting this knowledge from sentences on the web, it learns to extract this knowledge from semi-structured web data such as tables and lists, it learns morphological regularities of instances of categories, and it learns probabilistic horn clause rules that enable it to infer new instances of relations from other relation instances that it has already learned.

Overall, this project was a success in meeting its ambitious goals and deliverables. At the end of this project we now have successfully developed a system that has been running 24 hours/day, 7 days/week, for six weeks, and continues to run.

2. Technical Approach

This system performs continuous learning and continuous information extraction from the web. Each day it (1) extracts additional beliefs from 500,000,000 web pages in order to populate its growing knowledge base, and (2) learns to improve its reading ability, by learning new extraction patterns and new inference rules to infer yet more beliefs. As of February 22, 2010, this system has been running non-stop for over six weeks, processing half a billion web pages, and has extracted thus far approximately 200,000 beliefs to populate an ontology containing approximately 180 categories and relations. These categories range from "city" and "company" to "emotion" and "furniture", and relations range from "playsSport (athlete, team)" to "headquartedIn (company, city)." We evaluated the accuracy of the extracted facts when it had reached a collection of 88,000 beliefs, and at that point we found an estimated precision of 0.90. Importantly, NELL is designed as a never-ending language learner, and we intend to run it continuously for at least the coming year, funding permitting. Our estimate is that it will have well over 1 million extracted beliefs at that point, and an even more accurate learned reading ability.

The four key design choices in achieving this result are:

1. *A new approach to semi-supervised learning for information extraction*. In particular, whereas earlier work on semi-supervised learning for information extraction had primarily considered training a single extractor at a time (e.g., an extractor for "companies"), we developed in this project a much more accurate semi-supervised learning approach that learns hundreds of extractors simultaneously, and couples the training of all of these. By simultaneously learning these, and by coupling them using constraints available from the predicate ontology (eg., the constraints that "city" is a subset of "locations", is mutually exclusive with "emotions", and is a necessary condition for the second argument to the relation "mayorOf (politician, city)") we found we could achieve a major improvement in extraction accuracy.

2. *Designing learning methods whose accuracy automatically improves as the corpus size and ontology size scale up.* The semi-supervised learning algorithms noted above achieve an accuracy that improves as the number of ontology predicates increases. This is due to the fact that larger ontologies provide a larger nest of constraints to inform learning from unlabeled data. In addition, these algorithms *also* improve in accuracy as the size of the unlabeled text corpus grows. Therefore, we have increased our corpus to half a billion web pages collected by Prof. Jamie Callan (the ClueWeb data set), Carnegie Mellon University.

3. *Driving language processing to extract facts into a user-specified ontology.* In contrast to some other recent efforts at large scale information extraction, such as Prof. Oren Etzioni's "Textrunner" system (University of Washington), the NELL system requires the user to specify an initial ontology as input to NELL. This ontology defines the categories and relations of interest, and can be seen as equivalent to specifying a database schema. We find three advantages of this ontology-driven formulation of the problem, compared to ontology-free approaches such as Textrunner. First, the given ontology allows a prospective user to specify what types of information is of interest, as opposed to assuming every stated fact is of interest (note this is particularly important since we wish to grow the extracted knowledge base to fit the data schema for specific computer programs that need specific types of knowledge). Second, this ontology provides a focus for the language learning, so that the system can spend greater effort on the targeted knowledge types, initially ignoring thousands of other types of knowledge outside the ontology. Third, we find that the information provided as part of the ontology (e.g., that "city" is a subset of "locations" and is one of the argument types for "mayorOf") is precisely the kind of information used by our semi-supervised learner to couple the training of its multiple extractors and to produce its high accuracy. While we feel future work will need to examine how NELL can automatically extend the ontology initially provided by the user, the focus and constraints provided by this initial ontology are extremely important to the success of our approach.

4. *A learning architecture that attempts to learn multiple types of knowledge, where successful learning of one type of knowledge results in stronger learning of the other.* NELL currently learns four distinct types of knowledge. Of course, it is learning to extract many different ontology predicates (e.g., "city", "sportsTeam"), but here we mean to say that for each of these predicates it has four different learners that acquire four distinct and complementary ways to extract instances of the predicate. These are:

- Coupled Pattern Learner (CPL): A free-text extractor which learns and uses contextual patterns like "mayor of X" and "X plays for Y " to extract instances of categories and relations. CPL learns text extraction patterns, such as "if one finds the string 'mayor of X', then X is a city." CPL uses co-occurrence statistics between noun phrases and contextual patterns (both defined using part-of-speech tag sequences) to learn extraction patterns for each predicate of interest and then uses those patterns to find additional instances of each predicate. Relationships between predicates are used to filter out patterns that are too general. CPL is described in detail by Carlson et al. (2010). We used code provided by the authors. Probabilities of candidate instances extracted by CPL are heuristically assigned using the formula $1 - 0.5^c$, where *c* is the number of promoted patterns that extract a candidate. In our experiments, CPL was given as input a corpus of 2 billion sentences, which was generated by using the OpenNLP package (http://opennlp.sourceforge.net) to extract, tokenize, and part-of- speech tag sentences from the 500 million web page ClueWeb09 data set (Callan and Hoy 2009).

- Coupled SEAL (CSEAL): A semi-structured extractor which queries the Internet with sets of beliefs from each category or relation, and then mines lists and tables to extract novel instances of the corresponding predicate. CSEAL uses mutual exclusion relationships to provide negative examples, which are used to filter out overly general lists and tables. CSEAL is also described by Carlson et al. (2010), and we used code provided by the authors, based on that of Wang and Cohen (2009). Given a set of seed instances, CSEAL performs queries by sub-sampling beliefs from the knowledge base (KB) and using these sampled seeds in a query. CSEAL was configured to issue 5 queries for each category of interest and 10 queries for each relation of interest, and to fetch 50 web pages per query. Candidate facts extracted by CSEAL are assigned probabilities using the same method as for CPL, except that *c* is the number of unfiltered wrappers that extract an instance. CSEAL learns web page wrappers that typically capture Hyper-Text Markup Language (HTML) structure such as lists that support extraction.

- Coupled Morphological Learner (CML): A set of binary L2-regularized logistic regression models—one per category—which classify noun phrases based on various morphological features (words, capitalization, affixes, parts-of-speech, etc.). Beliefs from the KB are used as training instances, but at each iteration CML is restricted to predicates which have at least 100 positives. As with CSEAL, mutual exclusion relationships are used to identify negative instances. CML examines candidate facts proposed by other components, and classifies up to 30 new beliefs per predicate per iteration, with a minimum posterior probability of 0.75. These heuristic measures help to ensure high precision, generating increased support for existing candidates and enforcing morphological constraints on other subsystems. CML learns morphological classifiers of noun phrases that consider only the internal structure of the noun phrase to determine its type (e.g., it may learn that a noun phrase containing two capitalized words, where the second word ends in the three letters 'ski', is likely to be a person name).

- Rule Learner: learns probabilistic first order rules (horn clauses) that do not extract information from text, but instead capture empirical regularities among the hundreds of thousands of extracted beliefs, and infer new knowledge base beliefs directly from existing beliefs. For example, one learned rule indicates that "If AthletePlaysOnTeam (A,T) and TeamPlaysInLeague (T,L), Then AthletePlaysInLeague (A,L)"

Examples of knowledge acquired by each of these four learners are illustrated below.

Figure 1 shows example text extraction patterns acquired by CPL for the relation "AthletePlaysSport (arg1, arg2)." Note these patterns apply only if arg1 separately matches the definition of an "athlete" by the athlete classifier, and only if arg2 separately matches the definition of a "sport."

arg1_was_playing_arg2 arg2_megastar_arg1 arg2_icons_arg1 arg2_player_named_arg1
arg2_prodigy_arg1 arg1_is_the_tiger_woods_of_arg2 arg2_career_of_arg1 arg2_greats_as_arg1
arg1_plays_arg2 arg2_player_is_arg1 arg2_legends_arg1 arg1_announced_his_retirement_from_arg2
arg2_operations_chief_arg1 arg2_player_like_arg1 arg2_and_golfing_personalities_including_arg1
arg2_players_like_arg1 arg2_greats_like_arg1 arg2_players_are_steffi_graf_and_arg1 arg2_great_arg1
arg2_champ_arg1 arg2_greats_such_as_arg1 arg2_professionals_such_as_arg1
arg2_course_designed_by_arg1 arg2_hit_by_arg1 arg2_course_architects_including_arg1
arg2_greats_arg1 arg2_icon_arg1 arg2_stars_like_arg1 arg2_pros_like_arg1 arg1_retires_from_arg2
arg2_phenom_arg1 arg2_lesson_from_arg1 arg2_architects_robert_trent_jones_and_arg1
arg2_sensation_arg1 arg2_architects_like_arg1 arg2_pros_arg1 arg2_stars_venus_and_arg1
arg2_legends_arnold_palmer_and_arg1 arg2_hall_of_famer_arg1 arg2_racket_in_arg1
arg2_superstar_arg1 arg2_legend_arg1 arg2_legends_such_as_arg1 arg2_players_is_arg1
arg2_pro_arg1 arg2_player_was_arg1 arg2_god_arg1 arg2_idol_arg1 arg1_was_born_to_play_arg2
arg2_star_arg1 arg2_hero_arg1 arg2_course_architect_arg1 arg2_players_are_arg1
arg1_retired_from_professional_arg2 arg2_legends_as_arg1 arg2_autographed_by_arg1
arg2_related_quotations_spoken_by_arg1 arg2_courses_were_designed_by_arg1
arg2_player_since_arg1 arg2_match_between_arg1 arg2_course_was_designed_by_arg1
arg1_has_retired_from_arg2 arg2_player_arg1 arg1_can_hit_a_arg2 arg2_legends_including_arg1
arg2_player_than_arg1 arg2_legends_like_arg1 arg2_courses_designed_by_legends_arg1
arg2_player_of_all_time_is_arg1 arg2_fan_knows_arg1 arg1_learned_to_play_arg2
arg1_is_the_best_player_in_arg2 arg2_signed_by_arg1 arg2_champion_arg1

Figure 1: Example text extraction patterns

Table 1 shows example URL-specific extraction templates learned by CSEAL for a variety of ontology predicates. Note each row describes a different extraction pattern. The left column indicates the predicate being extracted, the second column the URL to which the patterns applies, and the third column gives the learned pattern.

Table 1: Example Text Extraction Patterns

Predicate	Web URL	Extraction Template
academicField	http://scholendow.ais.msu.edu/student/ScholSearch.Asp	` [X] -`
athlete	http://www.quotes-search.com/d_occupation.aspx?o=+athlete	`<a href='d_author.aspx?a=[X]'>-`
bird	http://www.michaelforsberg.com/stock.html	`<option>[X]</option>`
bookAuthor	http://lifebehindthecurve.com/	`</li> <li>[X] by [Y] &#8211;`

Table 2 shows weights learned by the morphological extractor CML for several ontology predicates. Positive and negative weights indicate positive and negative impacts on predicted probabilities, respectively. Note that “mountain” and “college” have different weights when they begin or end an instance. The learned model uses part-of-speech features to identify typical music group names (e.g., The Beatles, The Ramones), as well as prefixes to disambiguate art movements from, say, academic fields and religions.

Table 2: Example feature weights induced by the morphology classifier

Predicate	Feature	Weight
mountain	LAST=peak	1.791
mountain	LAST=mountain	1.093
mountain	FIRST=mountain	-0.875
musicArtist	LAST=band	1.853
musicArtist	POS=DT_NNS	1.412
musicArtist	POS=DT_JJ_NN	-0.807
newspaper	LAST=sun	1.330
newspaper	LAST=university	-0.318
newspaper	POS=NN_NNS	-0.798
university	LAST=college	2.076
university	PREFIX=uc	1.999
university	LAST=state	1.992
university	LAST=university	1.745
university	FIRST=college	-1.381
visualArtMovement	SUFFIX=ism	1.282
visualArtMovement	PREFIX=journ	-0.234
visualArtMovement	PREFIX=budd	-0.253

Table 3 shows first order horn clause rules acquired by RL. Each row describes a different rule, in which the consequent is concluded if the antecedents are satisfied. The probability in the left column is the probability of the consequent given the antecedents.

Table 3: Web page extraction templates learned by the CSEAL system

Probability	Consequent	Antecedents
0.95	athletePlaysSport(X, basketball)	$\Leftarrow$ athleteInLeague(X, NBA)
0.91	teamPlaysInLeague(X, NHL)	$\Leftarrow$ teamWonTrophy(X, Stanley Cup)
0.90	athleteInLeague(X, Y)	$\Leftarrow$ athletePlaysForTeam(X, Z), teamPlaysInLeague(Z, Y)
0.88	cityInState(X, Y)	$\Leftarrow$ cityCapitalOfState(X, Y), cityInCountry(X, USA)
† 0.62	newspaperInCity(X, New York)	$\Leftarrow$ companyEconomicSector(X, media), generalizations(X, blog)

These four learning methods are integrated using the system architecture in Figure 2 and summarized below. Notice the four “Subsystem components” in this diagram correspond to the four learning methods described above. Each learning method can access the shared knowledge base, and each can suggest new “candidate facts” to add to the knowledge base. The “Knowledge Integrator” component assesses the level of support for each candidate and determines which candidate facts to promote to full status as “beliefs” in the knowledge base. The ongoing run of the system is an iterative process in which each of these four modules can continually access the shared Knowledge Base to obtain new facts contributed by other modules, use these as new system-labeled training examples, retrain themselves and then propose new candidate facts.

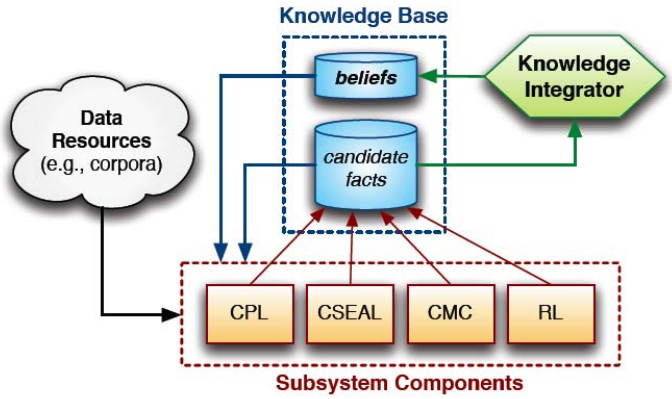

Figure 2: Never-Ending Language Learner Architecture

Our approach is organized around a shared knowledge base (KB) that is incrementally and continuously grown and used by a collection of learning/reading subsystem components that implement complementary knowledge extraction methods. The starting KB defines an ontology (a collection of predicates defining categories and relations), and a handful of seed examples for each predicate in this ontology (e.g., a dozen example cities). The goal of our approach is to continuously grow this KB by reading, and to learn to read better.

Category and relation instances added to the KB are partitioned into candidate facts and beliefs. The subsystem components can read from the KB and consult other external resources (e.g., corpora or the Internet), and then propose new candidate facts. Components supply a probability for each candidate and a summary of the source evidence supporting it. The Knowledge Integrator (KI) examines these candidate facts and promotes the most strongly supported of these to belief status. This flow of processing is depicted in Figure 2.

In our initial implementation, our approach operates iteratively. On each iteration, subsystem components are run to completion given the current KB, and then the KI makes its decisions on which candidate facts to promote. The KB grows, and this provides stronger training information to each component, and this in turn allows each component to learn to read better. In this way, our approach can be seen as implementing a coupled, semi-supervised learning method in which multiple components learn and share complementary types of knowledge, overseen by the KI.

This kind of iterative learning approach can suffer if labeling errors accumulate. To help mitigate this issue, the system may interact with a human for 10-15 minutes each day, to help the learner stay “on track,” though the experiments reported here make limited use of such human input.

The following design principles are important in implementing this approach:

- Use subsystem components that make uncorrelated errors. When multiple components that make uncorrelated errors propose the same candidate fact, we can typically be quite confident in that belief.

- Learn multiple types of inter-related knowledge. For example, we use one component that learns to extract predicate instances from text resources, and another which learns to infer relation instances from other beliefs in the KB. This provides multiple, independent sources of the same types of beliefs.

- Use coupled semi-supervised learning methods to leverage constraints between predicates being learned (Carlson et al. 2010). To provide opportunities for coupling, arrange categories and relations into taxonomy and declare most categories and relations to be mutually exclusive. Additionally, specify the expected category of each relation argument to enable type-checking. Subsystem components and the KI can benefit from methods that leverage coupling.

- Distinguish high-confidence beliefs in the KB from lower-confidence candidates, and retain source justifications for each belief.

- Use a uniform KB representation to capture candidate facts and promoted beliefs of all types, and use associated inference and learning mechanisms that can operate on this shared representation.

The implementation of the Knowledge Integrator (KI) promotes candidate facts suggested by the other components to the status of beliefs, using a hard-coded, intuitive strategy. Candidate facts that have high-confidence from a single source (those with posterior > 0.9) are promoted, and lower-confidence candidates are promoted if they have been proposed by multiple sources independently. KI exploits relationships between predicates by respecting mutual exclusion and type checking information. In particular, candidate category instances are not promoted if they already belong to a mutually exclusive category, and relation instances are not promoted unless their arguments are at least candidates for the appropriate category types (and are not already believed to be instances of a mutually exclusive category). In our current implementation, once a candidate fact is promoted as a belief, it is never demoted. The KI is configured to promote up to 250 instances per predicate per iteration, but this threshold was rarely hit in our experiments.

The KB in NELL is a reimplementation of the Theo frame-based representation (Mitchell et al. 1991) which was originally designed to support integrated representation, inference and learning.

3. Evaluation

An experimental evaluation was conducted to explore the following questions:

- Can NELL learn to populate many different categories (100+) and relations (50+) for 20+ iterations of learning and maintain high precision?

- How much do the different components contribute to the promoted beliefs held by NELL?

4. Methodology

The input ontology used in the experiments included 123 categories each with 10–15 seed instances and 5 seed patterns for CPL. Categories included locations (e.g., mountains, lakes, cities, museums), people (e.g., scientists, writers, politicians, musicians), animals (e.g., reptiles, birds, mammals), organizations (e.g., companies, universities, web sites, sports teams), and others. Fifty-five relations were included, also with 10–15 seed instances and 5 negative instances each (typically generated by permuting the arguments of seed instances). Relations captured relationships between the different categories (e.g., teamPlaysSport, bookWriter, companyProducesProduct).

In our experiments, CPL, CSEAL, and CML ran once per iteration. RL was run after each batch of 10 iterations, and the proposed output rules were filtered by a human. Manual approval of these rules took only a few minutes.

To estimate the precision of the beliefs in the KB produced by NELL, beliefs from the final KB were randomly sampled and evaluated by several human judges. Cases of disagreement were discussed in detail, with final decisions made by another judge. Facts which were once true but are not currently (e.g., a former coach of a sports team) were considered to be correct for this evaluation, as NELL does not currently deal with temporal scope in its beliefs. Spurious adjectives (e.g., “today’s Chicago Tribune”) were allowed, but rare.

5. Results

After six days of running, NELL completed 22 iterations of execution. 88,502 beliefs were promoted across all predicates (95% of these belonged to categories and 5% to relations.) Following an initial burst of almost 10,000 beliefs promoted during the first iteration, NELL continued to promote a few thousand more on every successive iteration, indicating strong potential to learn much more if it were left to run for a longer time.

To estimate the overall precision of these 88,502 beliefs, we sampled 100 of them uniformly and judged their correctness. 90 out of 100 were judged to be correct. Only a few items were debated by the judges: examples are “right posterior,” which was judged to not refer to a body part, and “green leafy salad,” which was judged acceptable as a type of vegetable. “Proceedings” was promoted as a publication, which we considered incorrect (it was most likely due to noun-phrase segmentation errors within CPL). Two errors were due to languages (“Klingon Language” and “Mandarin Chinese language”) being promoted as ethnic groups. (“Southwest”, “San Diego”) was labeled as an incorrect instance of the hasOfficesIn relation, since Southwest Airlines does not have an official corporate office there. Many system errors were subtle; one might expect a non-native reader of English to make similar mistakes.

To estimate precision at the predicate level, we randomly chose 7 categories and 7 relations which had at least 10 promoted instances for manual judgment. For each chosen predicate, we sampled 25 beliefs and judged their correctness. Table 4 shows these predicates, the estimates of precision, and the number of beliefs promoted in total for each. Most predicates are very accurate, with precision exceeding 90%. Two predicates, cardGame and productType, fare much worse. The cardGame category seems to suffer from the abundance of web spam related to casino and card games, which results in parsing errors and other problems.

As a result of this noise, NELL ends up extracting strings of adjectives and nouns like “deposit casino bonuses free online list” as incorrect instances of cardGame. Most errors for the productType relation came from associating product names with more general nouns that are somehow related to the product but do not correctly indicate what kind of thing the product is, e.g., (“Microsoft Office”, “PC”). Some of these productType beliefs were debated by the judges, but were ultimately labeled incorrect, e.g., (“Photoshop”, “graphics”). In our ontology, the category for the second argument of productType is a general “item” super-category in the hierarchy; we posit that a more specific “product type” category might lead to more restrictive type checking.

Table 4: Estimates of precision and numbers of promoted beliefs for selected predicates

Predicate	Precision	Promotions
cardGame	40%	584
city	92%	4311
magazine	96%	1235
recordLabel	100%	1384
restaurant	96%	242
scientist	96%	768
vertebrate	100%	1196
athletePlaysForTeam	100%	113
ceoOfCompany	100%	82
coachesTeam	100%	196
productType	28%	35
teamPlaysAgainstTeam	96%	283
teamPlaysSport	100%	79
teamWonTrophy	88%	119

As described in the technical section, NELL uses a Knowledge Integrator which promotes high-confidence single-source candidate facts, as well as candidate facts with multiple lower-confidence sources. CPL and CSEAL each were responsible for many promoted beliefs on their own. However, more than half of the beliefs promoted by KI were based on multiple sources of evidence. While RL was not responsible for many promoted beliefs, those that it did propose with high confidence appear to be largely independent from those of the other components.

CPL was designed to allow efficient learning of many predicates simultaneously from a large corpus of sentences extracted from web text. Gathering the statistics needed from the text corpus is the most expensive part of the algorithm. The statistics needed come from two types of queries. First, in the extraction step, CPL has a list of promoted instances and patterns, and needs to know which patterns and instances co-occur with those instances and patterns. Second, in the filtering and ranking steps, CPL needs to know which candidate patterns occur with which promoted instances, and which candidate instances occur with which promoted patterns. CPL gathers these statistics from a pre-processed text corpus which specifies how many times each noun phrase occurs with each category pattern in the corpus, and also how many times each pair of noun phrases occurs with each relation pattern. The preprocessing can be done quickly using the MapReduce framework (Dean and Ghemawat 2008). In each iteration of CPL, CPL gathers corpus statistics from this data set by scanning through the preprocessed data in two passes: one for extracting candidates and one for counting co-occurrences. CPL can perform one pass in about 15 minutes from a data set derived from 200 million web pages.

As for the quality of RL's learned rules, at iteration 10, RL proposed 85 rules, of which 75 (88%) were approved. At iteration 20, RL proposed 135 rules, of which 127 (94%) were approved.

6. Discussion

These results are promising. Preliminary implementation of NELL was able to maintain high precision and a consistent rate of knowledge accumulation with a very limited amount of human guidance. We consider this to be significant progress toward our goal of building a never-ending language learner.

The importance of our design principle of using components which make mostly independent errors is generally supported by the results. More than half of the beliefs were promoted based on evidence from multiple sources. However, in looking at errors made by the system, it is clear that CPL and CMC are not perfectly uncorrelated in their errors

This behavior suggests an opportunity for leveraging more human interaction in the learning process. Currently, such interaction is limited to approving or rejecting inference rules proposed by RL. However, we plan to explore other forms of human supervision, limited to approximately 10–15 minutes per day. In particular, active learning holds much promise by allowing NELL to ask “queries” about its beliefs, theories, or even features about which it is uncertain.

7. Conclusion and Ideas for the Future

We have developed architecture for a never-ending language learning agent, and described a partial implementation of that architecture which uses four subsystem components that learn to extract knowledge in complimentary ways. After running for six days, this implementation populated a knowledge base with over 88,000 facts with an estimated precision of 90%.

These results illustrate the benefits of using a diverse set of knowledge extraction methods which are amenable to learning, and a knowledge base which allows the storage of candidate facts as well as confident beliefs. There are many opportunities for improvement, though, including: (1) self-reflection to decide what to do next, (2) more effective use of 10–15 minutes of daily human interaction, (3) discovery of new predicates to learn, (4) learning additional types of knowledge about language, (5) entity-level (rather than string-level) modeling, and (6) more sophisticated probabilistic modeling throughout the implementation.

8. Publications associated with this project:

The publications below are available at Carnegie Mellon University's Read the Web project website, http://rtw.ml.cmu.edu/readtheweb.html.

- Coupled Semi-Supervised Learning for Information Extraction. Andrew Carlson, Justin Betteridge, Richard C. Wang, Estevam R. Hruschka Jr. and Tom M. Mitchell. *Proceedings of the Third ACM International Conference on Web Search and Data Mining (WSDM 2010).*

- Populating the Semantic Web by Macro-Reading Internet Text. Tom M. Mitchell, Justin Betteridge, Andrew Carlson, Estevam Hruschka, and Richard Wang. Invited paper, *Proceedings of the 8th International Semantic Web Conference (ISWC 2009).*

- Coupling Semi-Supervised Learning of Categories and Relations Andrew Carlson, Justin Betteridge, Estevam R. Hruschka Jr. and Tom M. Mitchell. *Proceedings of the NAACL HLT 2009 Workshop on Semi-supervised Learning for Natural Language Processing.*

9. References

Banko, M., and Etzioni, O. 2007. Strategies for lifelong knowledge extraction from the web. In K-CAP, 95–102.

Bunescu, R. C., and Mooney, R. J. 2007. Learning to extract relations from the web using minimal supervision. In Proc. of ACL, 576–583.

Callan, J., and Hoy, M. 2009. Clueweb09 data set. http://boston.lti.cs.cmu.edu/Data/clueweb09/.

Carlson, A.; Betteridge, J.; Wang, R. C.; Jr., E. R. H.; and Mitchell, T. M. 2010. Coupled semi-supervised learning for information extraction. In Proc. of WSDM.

Jeffrey Dean and Sanjay Ghemawat. MapReduce: simplified data processing on large clusters. Commun. ACM, 51(1):107-113, 2008.

Lenat, D. B. 1983. Eurisko: A program that learns new heuristics and domain concepts. AI Magazine 21(1-2):61–98.

Mitchell, T. M.; Allen, J.; Chalasani, P.; Cheng, J.; Etzioni, O.; Ringuette,M. N.; and Schlimmer, J. C. 1991. Theo: A framework for self-improving systems. Architectures for Intelligence 323– 356.

Nahm, U. Y., and Mooney, R. J. 2000. A mutually beneficial integration of data mining and information extraction. In Proc. of AAAI, 627–632.

Nii, H. 1986. Blackboard application systems and a knowledge engineering perspective. AI Magazine 7(3):82–107.

Quinlan, J. R., and Cameron-Jones, R. M. 1993. Foil: A midterm report. In ECML.

Wang, R. C., and Cohen,W.W. 2009. Character-level analysis of semi-structured documents for set expansion. In Proc. of EMNLP.

10. List of Acronyms

CPL	Coupled Pattern Learner
CML	Coupled Morphological Learner
HTML	Hyper-Text Markup Language
KB	Knowledge Base
KI	Knowledge Integrator
NELL	Never-Ending Language Learner
CSEAL	Coupled Set Expander for Any Language
RL	Rule Learner